Javascript föreläsning – Fredrik Thrlin

HTML – Struktur, ett skal

CSS – Färg/form med statiskt

JavaScript – Dynamisk. Bygger mycket på event tex ett click ”eventlistener” triggar igång en funktion. Mer levande och interaktiv.

HTML, CSS och JavaScript tolkas och körs av din webbläsare. Resultatet är en dynamisk hemsida.

**Historia:**

* Skapades 1995, Mocha, LiveScript, JavaScript
* ECMAScript (1997 – nu) Har skrivit en standard för hur webbläsaren ska se ut och det följer JacaScript. Det är standard i alla webbläsare.
* JavaScript !== Java (JavaScript och Java är två helt olika kodspråk)
* JavaScript serverside? Node.js Nu kan det användas som backendspråk, det har man inte kunnat tidigare. Client = Frontend Server = Backend
* ES 5 vs ES 6 Det försvinner inte saker, men det kan komma bya och bättre sätt att göra saker på.

Ligger längst ner innan </body> Det finns sätt att lägga den uppe i <head> Koden läs uppifrån och ner. Den läser in alla element först.

<script>

Javascript kod inuti script taggen

</script>

Vill helst skriva ut scripten såhär

<script src=”./script.js” />

**Syntax**

* Variabler (En behållare)
* Funktioner

**Värden:** Det är bara information som man jobbar med. Själva data.

- Fasta (Literals) 1337. ”Tomat” true

- Varierande (Variables) var myName = ”David”

**Nyckelord:**

* var =Så har man alltid skrivit variabler. Men det gör man inte nu. Har ersatts med två andra ord **let** eller **const** är det nya som man använder.
* **Let = Värdet kommer få ett nytt värde**
* **Const (constant) = värdet kommer aldrig ändra sig.**
* Function = Skapa eller deklarera en funktion
* Return = Används i funktioner, antingen att funktionen ska spotta ut ett värde som man kan ta hand om, eller att man vill hoppa ur en funktion.
* if else = Vilkor och logisk programering, man kan gå olika vägar.
* for while = Loopar tex om man har en lista, om man vill använda det
* switch = Gör man en lång if else, kan man ersätta det med en switch.
* case brake = Tillhär switch
* class =
* import = Man kan lägga kod i olika filer och importera in i andra filer. Mer moduler.
* try catch = Fel hantering

**Primitiva datatyper (Primitives) Oftast bara ett värde**

**String** – ”Tomat” en sträng dubbel eller enkel fnuttar ”” ´´ `` funkar lika bra. Om man vill skriva ut en text

**Number** – 1337, ett nummer JavaScript bryr sig inte om heltal eller halvtal

**Bigint** – Om man jobbar med tal som är extrem stora.

**Boolean** – True eller false det kan inte vara något annat. Finns bara dessa två alternativ.

**Symbol** -

**Undefined** – Det är om det finns en variabel. Men om man har missat att skriva.

**Null** – Det finns ingenting. Om man försöker titta på någonting som inte finns. Då kommer man få Null

JavaScript är ett löst skrivet språk. Typscript har mer regler där man måste typa sin kod, det är mer som andra kodspråk.

Operatorer

**Räkning:** + - \* / (Matematik)

**Tilldelning:** =

**Jämförelse:** == != <= >= (Oftast i if else)

== Lika med, om något är lika med något annat.

=== Strikt jämförelse, även datatypen

!= !== Inte är lika med något,

<= Om något är mindre än någonting

>= Om något är större eller lika med

**Uttryck** (1337 + 3) /2 ´Tomat´ + ´´ + ´är bra´

**Kommentarer** (Det är viktigt att skriva kommentarer, eller för att lösa en kod som inte fungerar, kommentera bort rader kod för att tex felsöka. Dokumentera eller skriva meddelanden till andra i ett projekt)

// Kommentar (En rad)

/\*Kommentar \*/ (Om man vill ha flera rader)

**Variabler (Se det som en låda med värden)**

var name, age, favoritePet, likesBeer;

name = ’Elon Musk’;

age = 22;

favoritePet = ’Dog’;

likesBeer = true:

console.log(name, age, favoritePet, likesBeer);

// Elon Musk 22 dog true <- Detta är vad som kommer skrivas ut i console.

Man kan använda det för felsökning, undersöka saker. Det är bara till för utvecklare.

**Variabler (Definition)**

Var, let, const = Det är detta som skapar lådan

En variabel är en storage location, en låda där man lägger saker. Spara en bit med data

Ge varje låda ett namn tex ->

const message

const message = ’’Hello world’;

consgt messade = ”Hello hello”;

console.log(’message’)

// Hello world

If, else

const weAreAwsome = true;

if (weAreAwsome) {

console.log(”Vi är otroligt bra!”;

}

Else {

console.log(”Hej”;

}

**Detta exempel är hårdkodat**

const day = ”Friday”;

if (day === ”Monday” {

console.log(”Yaay work”);

}

Else if (day === ”Friday”); {

Console.log(TGIF!);

}

Else if (day === ”Saturday” || day === ”Sunday”) {

Console.log(”Sweet sweet weekend”);

}

Else {

Console.log(”Work work”);

}

|| = Eller

&& = And

=== använder man oftast när man kodar

**Funktioner**

Man kan anropa funktioner i andra funktioner.

Funktionsdeklaration – Specifika funktioner myFunction men där körs inte för det

function myFunction () {

// Variabler som skapas här är bara synliga här

}

Funktionsanrop – Anrop och kör funktionen med namnet myFunction

myFunction();

function addTwoValues (firstValue, secondValue) {

return (firstValue + secondValue)

}

const addedValues = addTwoValues (1, 4);

querySelector = det är som CSS .classnamn #id

defer är en flagga som läggs i taggen om du vill ha ditt script i head. (Om det ska ligga längst upp. Annars går det bra att ha den längst ner.

Ett sätt att skriva strängar på. ${} för att ta in en variabel

`hej ${firstname}`;

alert(”Wllo world!”) Kommer upp som en pop-up ruta

prompt = det är en pop-up ruta med ett input fält.

Const result = prompt(”Hej hej”);

Console.log(result)

Event (button)

Funtction test () {

Console.log(”test”);

}

onClick=”test()”

querySelector = Hämtar in ett element baserat på #id .class

document.querySelector(”#leftBox”)

Math.floor(Math.random() \* 255; = ger oss ett slumpmässigt tal mellan 0-255 (Tex om man bygger ett spel och slår en tärning)